2T

УДК 004.771

М.Г. Ткаченко
НИИ многопроцессорных вычислительных систем им. академика А.В. Каляева
ЮФУ, г. Таганрог, Россия
itamm@mail.lviv.ua

Система обеспечения удаленного доступа с использованием Web-шлюза на платформе IIS/.Net/Silverlight

Разработана web-ориентированная система удаленного доступа с использованием шлюза прикладного уровня для обеспечения взаимодействия с ресурсами вычислительных сетей. Система предоставляет возможность удаленного управления как в рамках одной сети, так и при межсетевом взаимодействии. При этом специфика реализации шлюза в виде web-сервиса позволяет минимизировать ущерб защищенности вычислительной сети в целом.

Введение

Вопросы безопасности с распространением сетей публичного доступа приобрели особую актуальность. Существует большое разнообразие способов защиты компьютерных сетей, основанных как на использовании специализированного оборудования, так и на применении особых архитектур построения, использовании всевозможных межсетевых экранов, выделении демилитаризованных зон (DMZ) и многое другое.

Особое внимание при использовании различных средств защиты системные администраторы должны уделять не только ограничению доступа и защите информации, но также и обеспечению доступности сети для служебных целей. Кроме того, не менее важно обеспечивать максимально широкий доступ к сервисам изнутри локальной сети в условиях использования всевозможных прокси-серверов. Зачастую для решения этих задач приходится ослаблять защиту сети, чтобы обеспечить доступ к внутренним ресурсам снаружи, а широко используемый подход с организацией VPN-соединений может оказаться неприменимым [1], если со стороны удаленной сети разрешен доступ вовне только к сервисам WWW (по 80-му порту).

Основной причиной возникновения необходимости нестандартного доступа к ресурсам является системное администрирование, а именно удаленный доступ к «своим» серверам, как правило, скрытым в DMZ, из гостевой сети. Основным способом удаленного администрирования является удаленный терминал, в настоящее время наиболее широко распространены SSH (Secure Shell), Telnet (отходит на второй план ввиду совершенной незащищенности), VNC (Virtual Network Computing) и некоторые другие. Однако эти сервисы используют свои собственные TCP-порты для подключения, что требует организации в «гостевой» сети трансляции адресов и вообще, дополнительных накладных расходов для администратора сети.

В связи с этим, **представляется практически актуальной** разработка шлюза прикладного протокола, который позволяет, с одной стороны, использовать только один протокол доступа в сеть (а именно, HTTP), и при этом обеспечивает удаленное подключение к виртуальным терминалам. При такой организации доступ к серверу может быть получен с любого компьютера, или даже устройства мобильной связи, подключенного к сети Интернет, через стандартный Web-браузер.

Целью данной работы является создание системы обеспечения удаленного доступа через Web-шлюз. Разработанная система должна предоставлять доступ к удаленным компьютерам через Web-браузер.

При таком подходе администратору достаточно в своей сети организовать Web-сервер (что, как правило, делается почти с самого начала) и разместить на нем шлюзовое программное обеспечение. В дальнейшем для административного и пользовательского терминального доступа из практически любой «гостевой» сети достаточно иметь возможность использовать любой браузер и иметь доступ к «своему» Web-серверу.

Для обеспечения работы клиентов через Web-браузер **требуется решить следующие задачи**: на платформе Microsoft .Net разработать шлюз удаленного доступа, использующий пакет IIS (Internet Information Server). Шлюз обеспечивает взаимодействие между Web-браузером клиента и программой-сервером удаленного доступа. Взаимодействие с пользователем на клиентской стороне необходимо реализовать с применением технологии Microsoft Silverlight.

1. Схема подключения удаленного управления

Большинство систем удаленного доступа состоят из серверной части и средства просмотра (клиента удаленного доступа). Сервер устанавливается на компьютер, которым предполагается управлять. После инсталляции серверной части, администратор, запустив программу просмотра, сможет получить с ее помощью удаленный доступ к дисплею, клавиатуре и мыши сервера. Ниже приведен логический эквивалент организации клиент-серверной системы удаленного доступа (рис. 1).



Рисунок 1 – Диаграмма клиент-серверной системы удаленного доступа

Как видно из рисунка, компонентами подключения удаленного доступа являются клиент удаленного доступа, сервер удаленного доступа и инфраструктура коммуникационной сети (Wide area network, WAN) [2].

Почти все клиенты удаленного доступа, взаимодействующие на основе протокола PPP (Point-to-Point Protocol – протокол «точка-точка»), включая клиентов UNIX и Apple Macintosh, имеют возможность подключаться к серверу удаленного доступа под управлением Windows и других операционных систем.

Сервер удаленного доступа принимает подключения удаленного доступа и обеспечивает взаимодействие между клиентами удаленного доступа и оборудованием персонального компьютера, к которому он подключен.

Физическое или логическое подключение между сервером и клиентом удаленного доступа поддерживается с помощью оборудования, установленного на сервере и клиенте удаленного доступа, а также с помощью инфраструктуры телекоммуникаций. Тип оборудования для удаленного доступа и инфраструктура телекоммуникаций различаются в зависимости от типа устанавливаемого подключения.

Организованные таким образом системы удаленного доступа используют для взаимодействия свои собственные TCP-порты. Зачастую в компьютерных сетях для повышения защищенности устанавливается блокирующее оборудование, препятствующее использованию большинства TCP-портов, что может привести к невозможности обеспечения удаленного управления.

Рассмотрим упрошенный процесс обеспечения удаленного доступа. Взаимодействие компонентов удаленного доступа начинается с посылки клиентом пакета запроса на установление подключения, содержащего информацию об удаленном пользователе (в том числе имя пользователя и пароль), параметры подключения и, возможно, техническую информацию о клиентской программе.

Сервер, получив запрос, проверяет права доступа для удаленного клиента и согласовывает с клиентским программным обеспечением параметры подключения.

При успешном согласовании параметров подключения начинается процесс удаленного управления. Не вникая в подробности, можно описать этот процесс как последовательное выполнение следующих действий:

– клиент удаленного доступа отслеживает действия пользователя (такие как движения и клики мышью, нажатия клавиш) и упаковывает их описание в специализированные пакеты, формат которых определяется протоколом взаимодействия;
– сформированные пакеты отправляются серверу удаленного доступа;
– получив соответствующий пакет, серверная программа извлекает описание действий удаленного пользователя и воспроизводит их на серверном компьютере;
– в случае если действия пользователя привели к изменению состояния компьютера (зачастую отслеживается изображение рабочего стола компьютера), сервер формирует описание изменений и отправляет это описание клиентской программе;
– получив описание изменений, клиентская программа обновляет собственное состояние, эмулируя, таким образом, состояние удаленного компьютера.

Описанная последовательность передачи запросов при удаленном управлении представлена ниже (рис. 2).

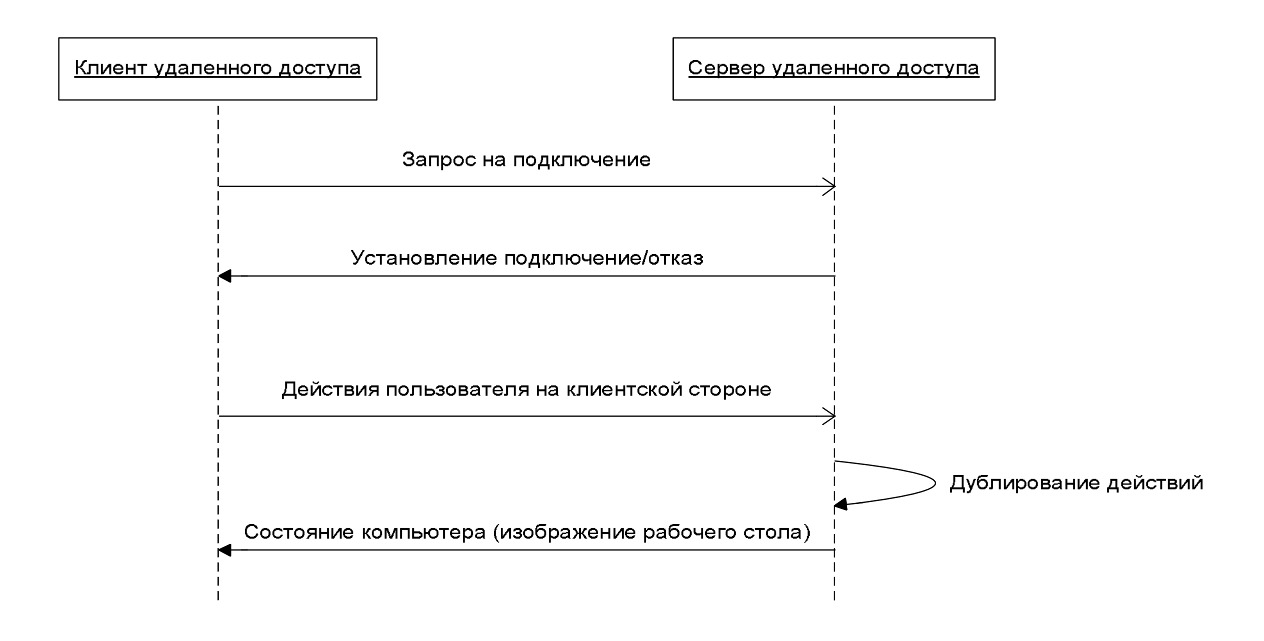


Рисунок 2 – Передача запросов при удаленном управлении

Циклическое выполнение описанных этапов позволяет обеспечить удаленное управление компьютером.

Следует отметить, что реализация последовательности и структуры этапов может отличаться в различных реализациях систем удаленного доступа.

Используя данное описание, можно разработать модифицированную архитектуру с использованием прикладного шлюза для доступа к серверу удаленного доступа.

2. Подключение с использованием Web-шлюза

Очевидно, что при использовании в качестве клиента удаленного доступа программного обеспечения Web-браузера требуется расширить описанную ранее архитектуру удаленного доступа компонентом, обеспечивающим взаимодействие между Web-браузером и сервером удаленного доступа – так называемым шлюзом.

Для снижения сложности и сокращения набора функций, возлагаемых на шлюз удаленного доступа, автор предлагает логически «разбить» его на два связанных компонента.

Первый компонент – посредник удаленного доступа – функционально приближен к клиенту удаленного доступа, описанному ранее. На него возлагаются функции протокольного взаимодействия с сервером удаленного доступа. С другой стороны, он связан со вторым компонентом шлюза – Web-сервером. С учетом специфики выполняемой задачи, посредника удаленного доступа целесообразно реализовать в виде Web-сервиса, к которому будет обращаться соответствующий Web-сервер.

На второй компонент – Web-сервер в классическом понимании, возлагаются функции обработки событий Web-браузера удаленного клиента, и передача их через сервис посредника удаленного доступа на сторону сервера удаленного доступа.
Описанные компоненты и связи между ними изображены ниже (рис. 3).

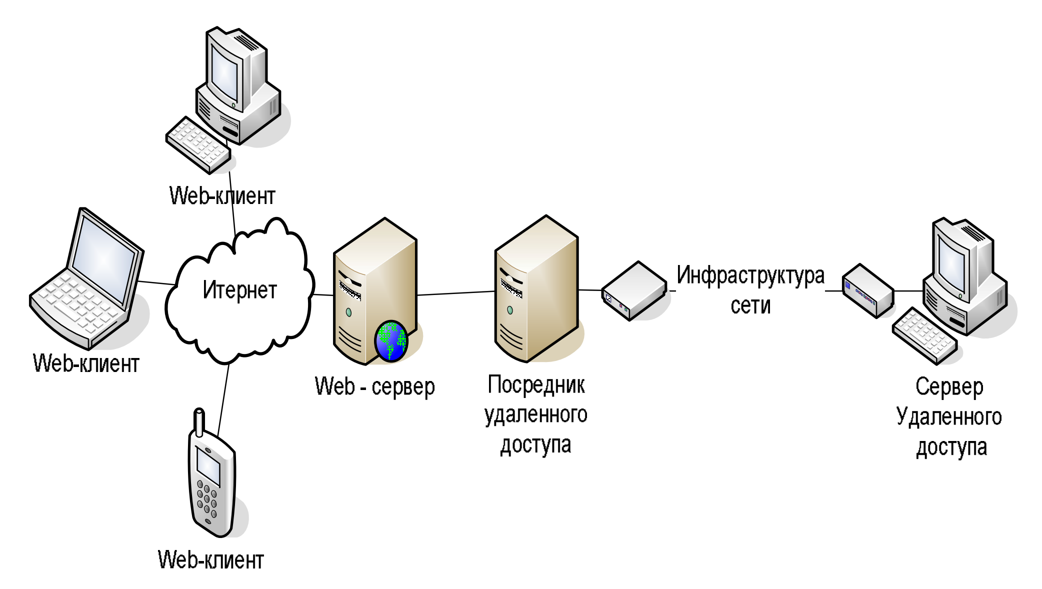


Рисунок 3 – Подключение удаленного управления через Интернет

Роль Web-клиентов в данном случае играют браузеры конечных пользователей, такие как Microsoft Internet Explorer или Mozilla Firefox. Единственным требованием, предъявляемым к браузеру, является наличие установленного расширения Microsoft

Silverlight для запуска компонентов удаленного управления. Эти расширения распространяются бесплатно и доступны для скачивания с сайта Microsoft(www.Microsoft.com/Silverlight).

В качестве Web-сервера используется пакет Microsoft Internet Information Services, поставляемый компанией Microsoft в комплекте с операционной системой Windows. Основной задачей Web-сервера является формирование Web-страниц и предоставление доступа к ним со стороны клиента.

Клиент-посредник удаленного доступа обеспечивает взаимодействие между Web-клиентами и сервером удаленного управления. Таким образом, основными задачами посредника являются получение данных с удаленного сервера, подготовка и передача их на сторону клиента, а также информирование удаленного сервера о действиях пользователя, таких как перемещения указателя мыши и нажатие клавиш. Последовательность передачи запросов между компонентами системы аналогична описанной ранее передаче запросов в клиент-серверной архитектуре (рис. 4).

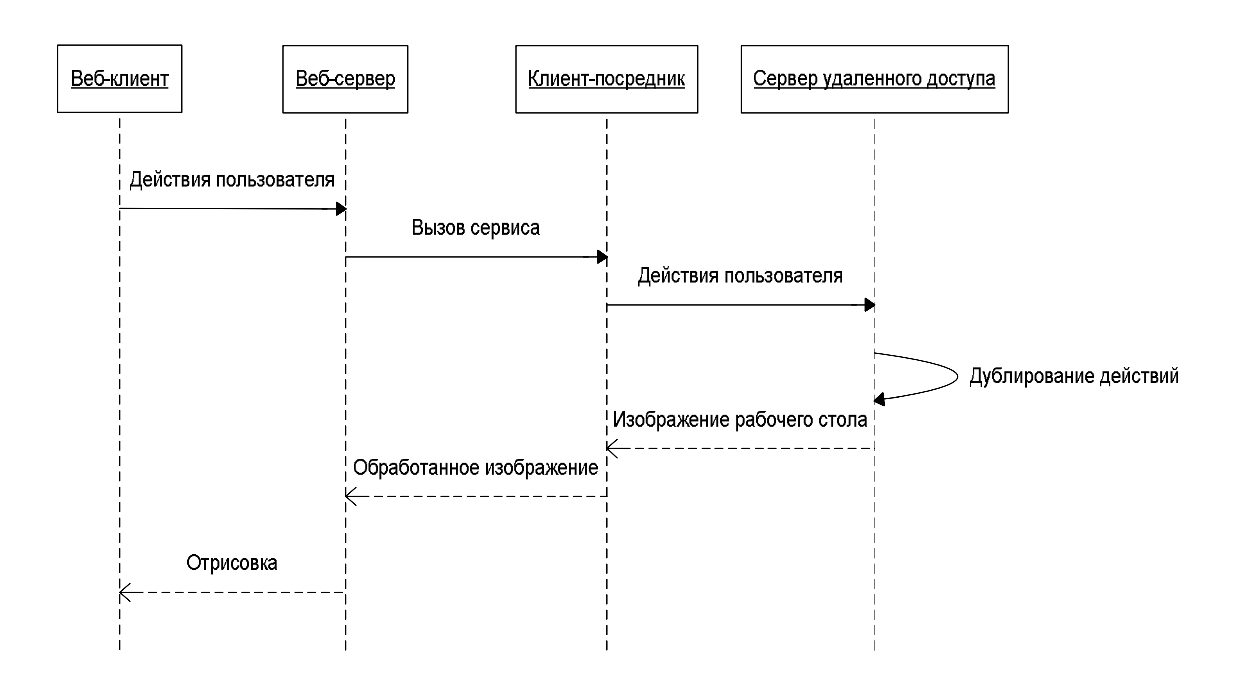


Рисунок 4 – Последовательность передачи запросов

Отличие архитектуры состоит в необходимости ретранслирования промежуточными компонентами запросов между Web-клиентом и сервером удаленного доступа.

Архитектура разработанной системы предполагает логическое разделение ее на ряд взаимосвязанных компонентов, к числу которых относятся:

1) сервер удаленного доступа;
2) посредник удаленного доступа;
3) web-сервер;
4) web-клиент.

Описание реализации предполагается начать с рассмотрения Web-клиента, после чего перейти последовательно к Web-серверу, посреднику удаленного доступа и закончить рассмотрение описанием сервера удаленного доступа.

3. Web-клиент удаленного доступа

В задачи Web-клиента входит обеспечение взаимодействия с конечным пользователем разработанной системы. Программная реализация Web-клиента создана с применением технологии Microsoft Silverlight.

Silverlight – это новая технология представления данных в Интернете, предназначенная для запуска на различных платформах. Она позволяет создавать визуально привлекательные Web-страницы, работающие в различных обозревателях, устройствах и настольных операционных системах [3]. Ключом к возможностям Silverlight, как и ко всей технологии представления WPF (Windows Presentation Foundation) платформы Microsoft .NET Framework 3.0, является XAML (eXtensible Application Markup Language, расширяемый язык разметки приложений).

Поскольку технически XAML – это XML, он представляет собой простой текст, а значит, не вызывает конфликтов с брандмауэрами, легко доступен для просмотра, и при этом описывает различное содержимое. Некоторые технологии – Java, ActiveX, Flash – в настоящее время широко применяются в дополнение к языкам DHTML, CSS и JavaScript и расширяют содержимое Web-страниц, но их роднит один недостаток – данные передаются в обозреватель в двоичном виде. Такую информацию сложно проверить на предмет безопасности, не говоря уже о сложности ее обновления – для реализации любых изменений требуется переустановка всего приложения, что неудобно для пользователя и зачастую приводит к торможению Web-страниц. При изменении содержимого страницы средствами Silverlight новый XAML-файл создается на стороне сервера. При следующем просмотре страницы происходит загрузка этого файла, а значит, потребность в переустановке отпадает.

Основой технологии Silverlight является модуль расширения для обозревателя, который обрабатывает XAML и отображает итоговое изображение в поле обозревателя. Загрузочный файл, содержащий модуль, может быть установлен при посещении пользователем узла с содержимым, создававшимся с использованием Silverlight. Модуль предоставляет разработчикам доступ к функциям XAML-страницы на языке C#, таким образом, становится возможным взаимодействие с содержимым на уровне страницы и разработчик может, например, создать обработчики событий или управлять содержимым XAML-страницы с помощью программного кода.

Модуль Web-клиента разработан с использованием объектно-ориентированного подхода.

Очевидно, что реализовать взаимодействие с сервером удаленного доступа напрямую из Web-клиента невозможно. Для реализации данного взаимодействия разработан специализированный Web-сервис, проксирующий передачу данных между Web-клиентом и сервером удаленного доступа.

Классы, входящие в состав Web-клиента, представляют собой оболочку для вызова методов Web-сервиса удаленного доступа. В первую очередь интерес представляет класс VNCRemoteServiceProviderSoapClient, представляющий собой обертку для вызовов функций Web-сервиса.

При вызове методов Web-сервиса были использованы асинхронные запросы. Например, обработчик срабатывания таймера выглядит следующим образом:

```
void RefreshTimer_Tick(object sender, EventArgs e)
{
 if (m_bRequestSent)
  return;
 if (m_bMouseStateChanged)
 {
  m_bRequestSent = true;
```

```
      m_bMouseStateChanged = false;
      AddLog("Timer: mouse event.");
      m_VNCProxy.MouseEventAsync(m_nValidatedMousePositionX,
m_nValidatedMousePositionY, m_bLeftMouseButtonPressed, false);
      return;
    }

    m_bRequestSent = true;
    AddLog("Timer: refresh window.");
    m_VNCProxy.GetRemoteWindowAsync();
 }
```

Методы GetRemoteWindowAsync и MouseEventAsync, использованные при написании этого кода, предполагают выполнение асинхронных запросов к Web-сервису. С учетом специфики выполнения асинхронных запросов требуется описать обработчики события завершения запроса. Например, обработчик завершения события MouseEvent выглядит так:

```
     void m_VNCProxy_MouseEventCompleted(object sender,
MouseEventCompletedEventArgs e)
   {
    AddLog("Mouse event complete.");
    m_bRequestSent = false;
   }
```

А также необходимо связать обработчики с соответствующими событиями:

```
  m_VNCProxy.GetRemoteWindowCompleted += new EventHandler
<GetRemoteWindowCompletedEventArgs> (VNCProxy_GetRemoteWindowCompleted);
  m_VNCProxy.MouseEventCompleted += new EventHandler
<MouseEventCompletedEventArgs>(m_VNCProxy_MouseEventCompleted);
```

Все методы созданного Web-сервиса вызывались асинхронно, в случае использования синхронного вызова используется только 1 функция, функции-обработчика завершения не будет.

Синхронный вызов проще в реализации, но он имеет значительный недостаток – приложение становится недоступным для взаимодействия с пользователем, оно «зависает» до получения ответа. Асинхронный вызов более удобен и гибок, и именно его советуют использовать в проектах [4].

4. Web-сервер

В задачи Web-сервера входит проверка наличия поддержки расширения Silverlight 2.0 со стороны пользовательского Web-браузера, а также загрузка на сторону пользователя .xap-файлов, содержащих описание сцен Silverlight.

Проверка поддержки пользовательским Web-браузером осуществляется с помощью шаблонных макросов, предоставляемых фирмой Microsoft с набором разработчика Silverlight.

Для включения сцен Silverlight HTML-страницы содержит вызов метода create-Silverlight(), находящегося в фоновом коде. Например, для страницы Default.html файл с фоновым кодом будет иметь название Default.html.js и содержать следующий текст:

```
Sys.Silverlight.createObjectEx({
  source: "Scene.xaml",
  parentElement: document.getElementById("SilverlightControlHost"),
```

```
  id: "SilverlightControl",
  properties: {
    width: "100%",
    height: "100%",
    version: "0.9"
  },
  events: {
    onLoad: Sys.Silverlight.createDelegate(scene, scene.handleLoad)
  }
 });
```

В этот метод передается ряд свойств, в том числе те, что используются для указания отображаемого XAML-кода, внешнего вида элемента управления Silverlight и обработчиков событий onLoad и onError.

Свойство source: используется для определения XAML, который нужно отобразить на странице. Это может быть внешний файл (как в нашем случае) или расположенный на странице именованный тег <script>, содержащий XAML-код.

Размещая элемент управления Silverlight на странице, нужно поместить его в именованный тег <DIV>. Свойству parentElement: следует присвоить имя этого тега <DIV>.

Идентификатор элемента управления указывается в свойстве id.

Физические характеристики – высота, ширина и версия – задаются с помощью массива, передаваемого свойству properties.

Загрузка стартовой страницы, содержащей описание Silverlight-проигрывателя презентационного видеоролика, начинается автоматически при загрузке стартовой страницы. Загрузка и открытие остальных страниц осуществляется в асинхронном режиме под управлением клиентского модуля. Использование асинхронных запросов позволяет избежать задержек при переходах между сценами. Так, например, в разработанной системе использована возможность загрузки XAML-файлов в процессе воспроизведения презентационного видеоряда. Это позволяет пользователю по завершении просмотра преступить к работе с системой, не ожидая загрузки новой сцены в Silverlight.

5. Посредник удаленного доступа

При разработке системы возникла задача передачи данных между Web-клиентом, использующим технологию Microsoft Silverlight, и сервером удаленного доступа, взаимодействующим через TCP подключение.

Очевидно, напрямую такое взаимодействие обеспечить невозможно. Для решения этой задачи был разработан Web-сервис, взаимодействующий с сервером удаленного доступа и имеющий точки доступа для вызова Web-клиентом.

Основным классом, предоставляющим точку доступа для Web-клиентов, является VNCRemoteServiceProvider.

Среди методов данного класса можно выделить ConnectToRemoteServer – метод, обеспечивающий подключение Web-клиента к серверу удаленного доступа:

```
   [WebMethod(EnableSession = true)]
   public bool ConnectToRemoteServer(string IPAddress, int Port, string Password)
   {  bool Success = false;
     Conn Connection = null;
     try
     {
  Connection = new Conn(IPAddress, Port, Password);
```

```
    if (!Connection.Run()) Connection.CleanUp();
    Session["ActiveConnection"] = Connection;
    Success = true;
  }
  catch(Exception ex)
  {  Debug.Assert(false, ex.Message);  }
  return Success;
}
```

В целом можно отметить, что объявление службы отличается от обычного метода только строчкой [WebMethod] и ограничениями на входные и возвращаемые типы данных. Web-служба может принимать и возвращать параметры следующих типов:
простые типы данных – строки, целые, числа с плавающей точкой;
массивы;
объекты – передаются все общедоступные свойства какого-либо объекта;
перечисления - типы в C#, определяемые ключевым словом enum;
XmlNode – представляют собой часть Xml документа;
DataSet, DataTable – применяются для возврата данных из БД для последующей привязки к элементам отображения данных в .NET. Не подходит для использования в Silverlight.

В процессе реализации посредника удаленного доступа оказалось невозможным передавать на сторону клиента получаемые изображения рабочего стола через вызов Web-методов. Для решения этой проблемы был спроектирован специализированный файловый буфер. При запросе клиентом изображения с рабочего стола сервис посредника переадресует запрос на сторону сервера удаленного доступа. При получении ответа от сервера, сервис сохраняет полученное изображение в виде графического файла и передает Web-клиенту в качестве результата запроса имя сохраненного файла.

Подобное решение имеет несколько преимуществ. Во-первых, уменьшаются задержки на передачу данных между клиентом и Web-сервисом, что в свою очередь уменьшает загруженность сервера. Во-вторых, появляется возможность уменьшить передаваемый трафик. Экономии можно добиться, если не передавать изображения с рабочего стола удаленного компьютера на сторону клиента в случае отсутствия каких-либо изменений в промежутке между запросами. При применении файлового буфера подобную проверку можно вынести в функционал Web-сервиса, при этом незначительно увеличив нагрузку на него.

6. Сервер удаленного доступа

В качестве сервера удаленного доступа использован разработанный компанией AT&T Laboratories VNC-сервер. Данный программный продукт распространяется бесплатно с открытыми исходными кодами. В соответствии с лицензией, проект доступен для распространения и изменения.

Проект компании AT&T Laboratories полностью реализован на языке C#, совместим на уровне двоичного кода со смартфонами, карманными персональными компьютерами и настольными компьютерами.

В качестве сервера удаленного доступа использован VNC-сервер, использующий протокол RFB версии 3.4.

Описание свойств, реализованного сторонним производителем сервера удаленного доступа, выходит за рамки данной работы. Подробную информацию о разработке можно получить на официальном сайте производителя [5].

Следует отметить, что реализация в программном коде посредника удаленного доступа полной функциональности протокола RFB делает возможным подключение не только к серверу, разработанному AT&T Laboratories, но и к любому VNC-серверу, использующему протокол RFB 3.4 или более ранних версий.

Заключение

Разработана система обеспечения удаленного доступа с использованием прикладного шлюза. Основное отличие от классического клиент-серверного взаимодействия заключается во введении дополнительных компонентов, обеспечивающих доступ со стороны программного обеспечения Web-браузера конечного пользователя к серверу удаленного доступа.

В разработанной системе предусмотрены возможности для дальнейшего развития, в том числе применения новых алгоритмов сжатия передаваемых данных.

Разработка может быть эффективно использована в самых различных предметных областях. Предполагается ее интеграция в архитектуру территориально распределенных мехатронных комплексов технологических объектов критических областей деятельности, таких как авиация, нефтедобывающая отрасль, медицина, операции на фондовом рынке.

Литература

1. Virtual Private Networks. – Режим доступа: http://www.microsoft.com/vpn.
2. Обзор программ для удаленного администрирования. – Режим доступа: http://www.1on.ru/2006_10_16/obzor.html
3. Silverlight developer center. – Режим доступа: http://msdn.microsoft.com/en-us/silverlight.
4. Управление длительными процессами ASP.Net+Web Service. – Режим доступа: http://gotdotnet.ru/LearnDontNet/AspNet/682.aspx.
5. Remote control software. – Режим доступа: http://www.uk.research.att.com/vnc.

М.Г. Ткаченко

Система забезпечення видаленого доступу з використанням Web-шлюзу на платформі IIS/.Net/Silverlight

Розроблена web-орієнтована система видаленого доступу з використанням шлюзу прикладного рівня для забезпечення взаємодії з ресурсами обчислювальних мереж. Система надає можливість видаленого керування як в рамках однієї мережі, так і при міжмережній взаємодії. При цьому специфіка реалізації шлюзу у вигляді web-сервісу дозволяє мінімізувати збиток захищеності обчислювальної мережі в цілому.

M.G. Tkachenko

Remote Control System Using Web-gateway Based on IIS/.Net/Silverlight Platform

Web-oriented remote control system using application level gateway for network interaction was developed. The system gives opportunity of remote control within the bounds of a network and internetworking. In addition specificity of gateway implementation as web service allows to minimize network security damage.

Статья поступила в редакцию 17.07.2008.